ECE521: Inference Algorithms and Machine Learning
Report of Assignment 1

Team Members:
Xi Chen (1002266013)
Yang Wu (1002495132)

Task 1: Training Size

Set k = 1 and we get a 2-column table that compares the number of validation errors correlating to different training size as follows:

Training Size N	Validation Errors
5	107
50	40
100	27
200	17
400	9
800	12

Table 1. Training Size Vs. the Number of Validation Errors

As the table shows, initially, as the training size N increases, the number of validation errors decreases, because more information about each validation case are gathered ($H(Y|X_N,X_{N-1}, \ldots, X_1) < H(Y|X_{N-1},X_{N-2}, \ldots, X_1)$, where H is the uncertainty in the information theory) when the training size increases. But the number of validation errors is reaching to the lowest value when the training size hits 400, and then it will bounce up if N continue increasing since too much irrelevant information starts to be calculated and the result is exaggerated. Thus, we think neither large nor small N is best, N will have the best effect on performance in an intermediate value.

Task 2: Overfitting and Underfitting

When we use the complete training set of N = 800, we get a 2-column table that shows the number of validation errors versus the values of k as follows:

K	Validation Errors
1	12
3	8
5	10
7	9
21	11
101	24
401	51

Table 2. The value of K vs. the Number of Validation Errors

As the table shows, the number of validation errors is small when k is small and it will increase as k tends to be large since the closest neighbors have relatively high correlation and the increasing value of k introduces some unrelated training samples, which will lead to high uncertainty. Thus, k has the best effect on performance when it is small.

Task 3: Linear Fit

When we set the learning rate = 0.005 and the number of epochs = 10000 in the gradient descent, we get the following figure to show the fitted line and original data.

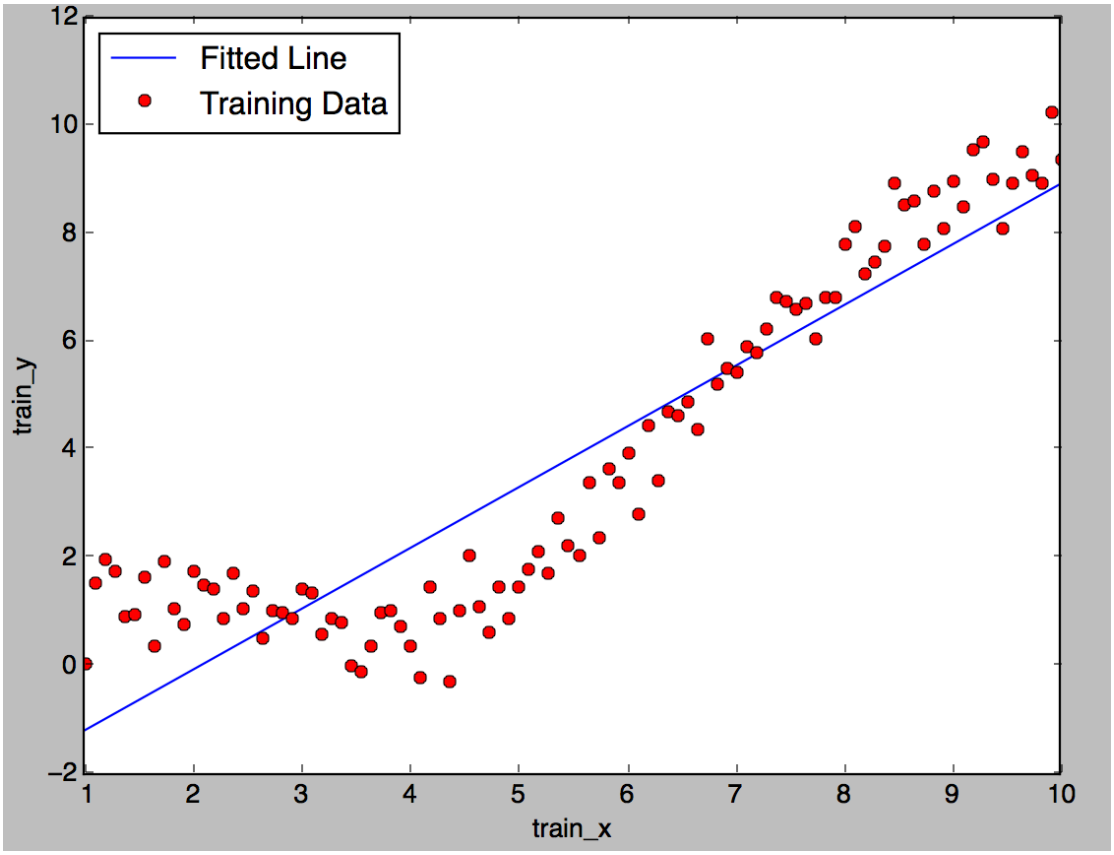


Figure 1. Training Set and Fitted Line

Task 4: Feature Space

When we map each input (train_x, train_y) up to higher dimensional feature spaces of $[1, x, x^2, x^3, x^4, x^5]$, and normalize each dimension of feature space by $x = \frac{x-\mu}{\sigma}$, we get the following figure that shows the training set and the fitted curve.

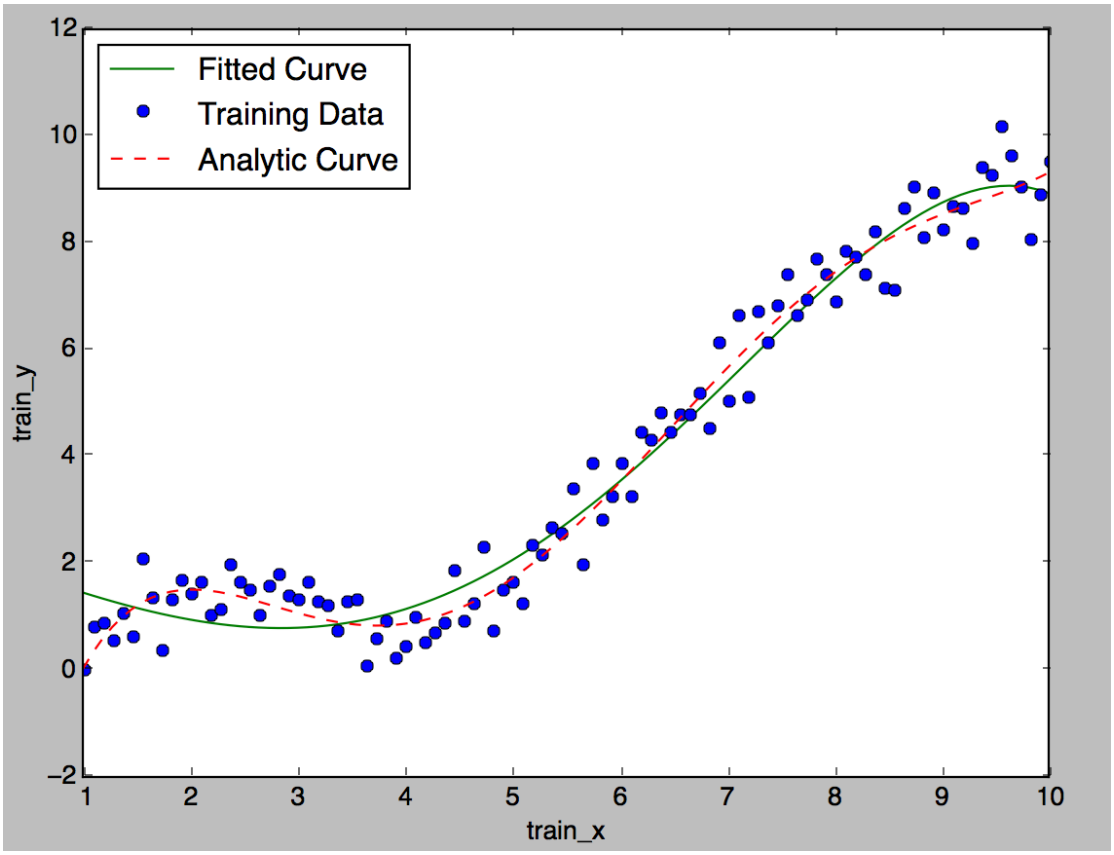


Figure 2. Training Data and Fitted Curve by non-linearity

By introducing non-linearity and high dimensional feature space, the fitted curve (green line) fits better than the linear regression algorithm in Task3. Yet, the analytic method of the same spatial dimension provides a better curve (red dash line).

Task 5: Training Size

We set the learning rate = 0.05 and the number of epochs = 2000 where $N \in [100, 200, 400, 800]$ in the stochastic gradient descent. We get the following 2-column table that displays the number of validation errors for each size N.

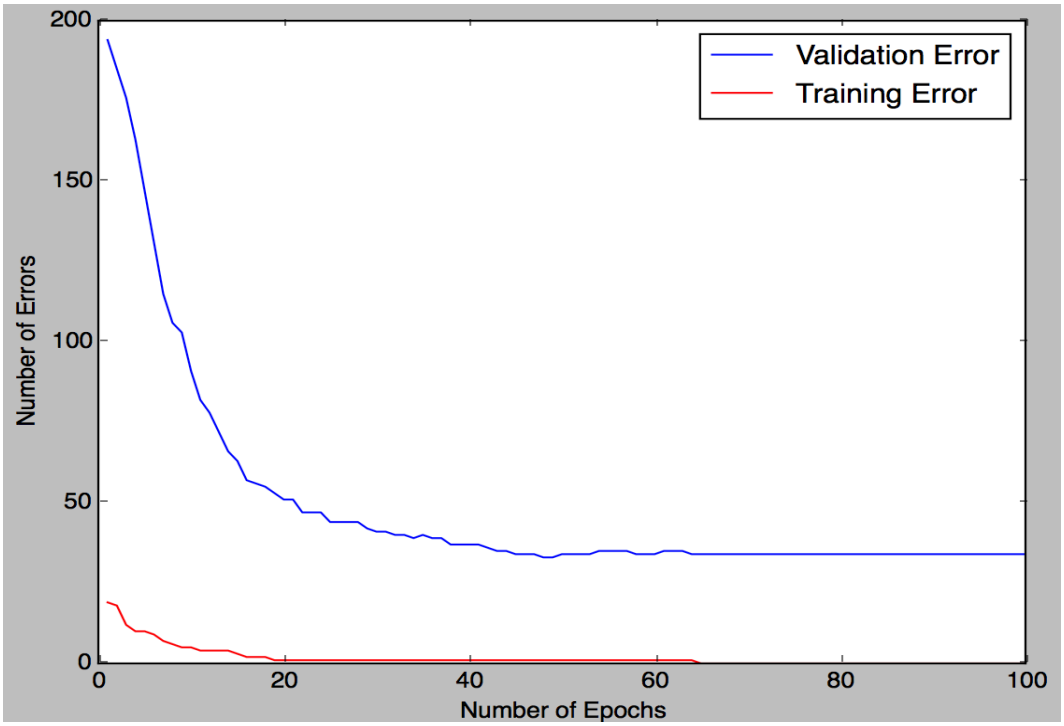
Training Size N	Validation Errors
100	35
200	26
400	25
800	23

Table 3. The Number of Validation Errors vs. Training Size N

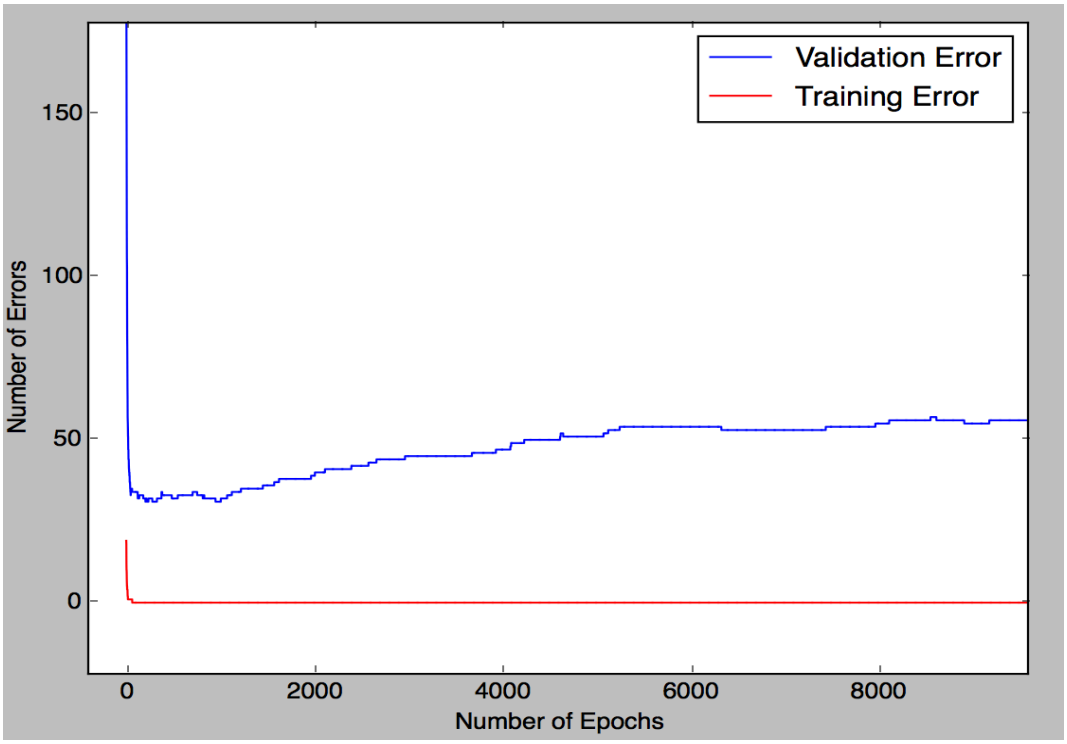
Here, we set the minibatch size = 50, so the weight is updated for 2000*N/50 times for each training size. As we can see in the table, the number of validation errors decreases when the training size N increases. Thus, we think the large training set N is optimal for our model.

Task 6: Overfitting

We set the learning rate = 0.05, the binary threshold = 0.5 and use the first N = 50 sample cases of the complete 800 cases as the training set. By using the stochastic gradient descent with different numbers of epochs, we get the following figure.



(a) 100 epochs in total



(b) 10000 epochs in total

Figure 3. The number of training errors and validation errors Vs. the number of Epochs

The graph has clearly shown that the training error will immediately drop to 0 after a few epochs, while the validation errors drop to 30 after 100 epochs. When epochs keep increasing, the validation errors will stay until the number of error bounces up again due to the overfitting.

Task 7: Regularization

We set the learning rate = 0.05 and train a linear regression with regularization parameters of $\lambda \in [0, 0.0001, 0.001, 0.01, 0.1, 0.5]$ using stochastic gradient descent in the first N = 50 cases of the complete dataset. We get the following 2-column tables that shows the number of validation errors for each λ.

λ	Validation Errors
0	39
0.0001	39
0.001	38
0.01	33
0.1	32
0.5	39

Table 4. The number of Validation Errors Vs. lambda

As the table 4 shows, increasing the value of λ to a certain value can improve the performance of the linear regression because it can reduce the impact of information with large weight w. However, it will lead to overcorrecting when the value of λ continues increasing since it will add bias to the estimation. Thus, increasing the value of λ can reduce the overfitting but lead to more bias. So, the best value of λ is 0.1 in our model.